Security: Certificates in Mac OS X

Session 114

Security: Certificates in Mac OS X

Craig Keithley
Security and Cryptography Technology Evangelist

Introduction

- Common Data Security Architecture
- New Security APIs in Mac OS X
- Keychain additions
- Certificate functionality

Security: Certificates in Mac OS X

Ken McLeod
Senior Software Engineer, Data Security

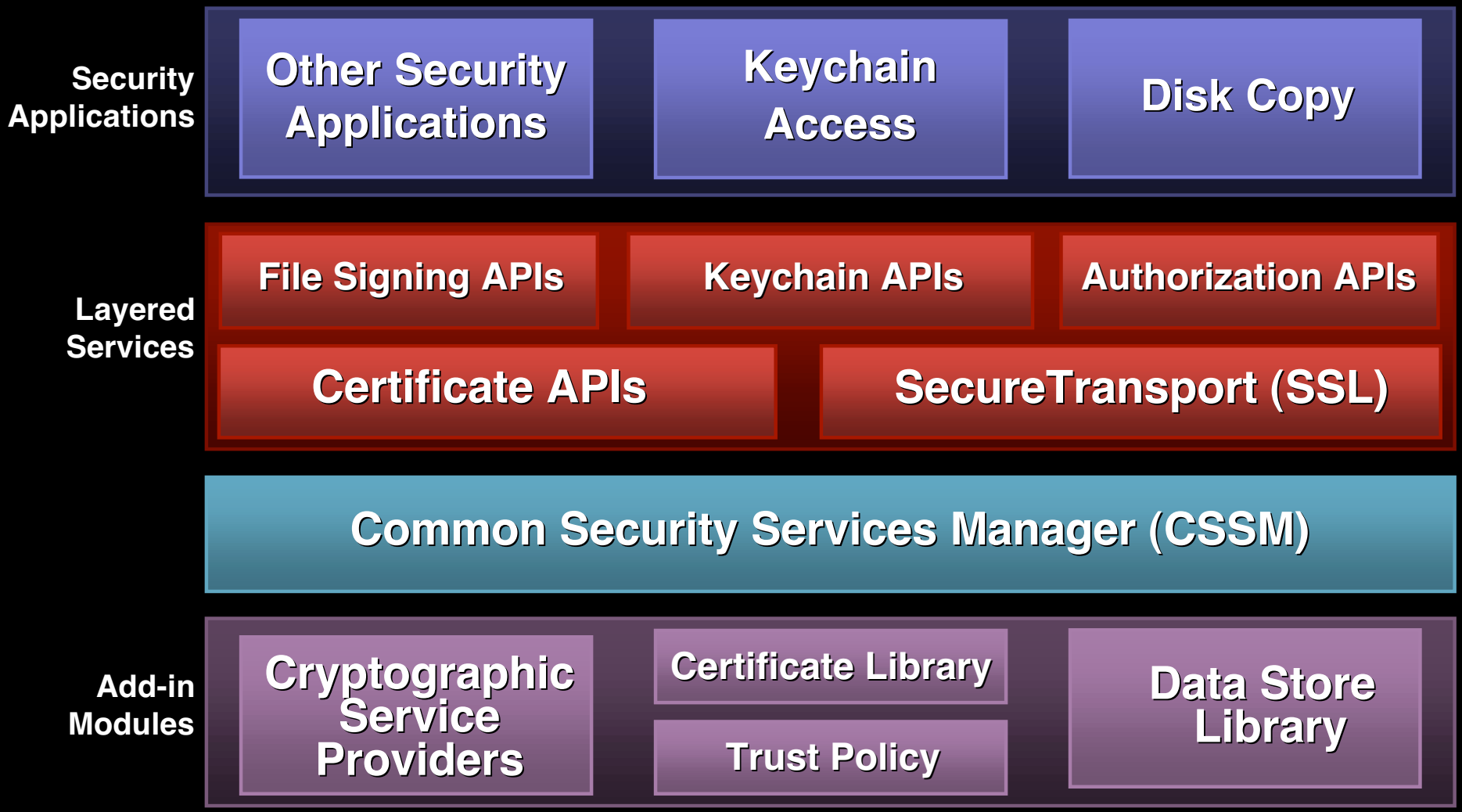
What You Will Learn

- Overview of CDSA
- Basic functionality provided by Security framework
- Using certificates to establish trust
- Using keychains to store certificates

CDSA Overview

- Modular plug-in architecture for security services
  - Cryptographic Service Provider (CSP)
  - Data Library (DL)
  - Certificate Library (CL)
  - Trust Policy (TP)
- Common Security Services Manager (CSSM) provides API access

CDSA Architecture



CDSA Functionality

- Key generation (symmetric and asymmetric)
- Encryption (AES, etc.)
- Digests (MD5, SHA-1)
- Data storage and retrieval
- Access control on keys
- Certificate parsing and evaluation

Security Framework

- Implements CDSA functionality in Mac OS X
- Provides CDSA interfaces
- Provides higher-level Security interfaces
- Part of Darwin (open source)
- User interface factored out for low-level access

About “Sec” APIs

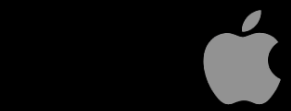
- Patterned after CoreFoundation
- In fact, they are CF objects
- Can use CFRetain, CFRelease, CFArray, etc.
- Be aware of Copy vs. Get
- SecObjectSearchRef searches for SecObjectRef
- Bridge functions to CSSM

SecKeychain

- Lower level than Keychain Manager APIs
- SecKeychainRef = KCRef = CFTypeRef
- May bring up UI (via SecurityAgent) unless user interaction is off
- More flexible; all attributes of a keychain item can be specified
- Access control support

X.509 Certificate Overview

- What's in a X.509 certificate?
  - Attributes (e.g., issuer, subject, validity)
  - Extensions (e.g., key usage, policies)
- Leaf certificates certify a public key
- Intermediate certificates certify others
- Root certificates certify themselves
- Used for SSL and S/MIME



Security: Certificates in Mac OS X

Perry “the Cynic” Kiehtreiber
Senior Software Engineer, Data Security

SecCertificate

- API for a certificate (X.509 for now)
- SecCertificateRef obtained from . . .
  - Plain data (CFData)
  - Keychain search (SecCertificateSearch)
  - API evaluation (e.g., SecTrust)

SecCertificate (Cont.)

- Used in APIs
  - SecureTransport (SSL)
  - Verification (SecTrust)
  - Identities (SecIdentity)
- Other uses
  - Organize in keychains
  - Establish user trust (SecTrust APIs)
  - Certificate display APIs
- No access control (do not panic!)

SecIdentity

- Represents a PKI *User Identity*
- Consists of Public Key Pair and Certificate
- Stored separately in keychains (virtual object)
- Access control through private key
- Good advice
    - Let user choose
    - Rely on system for access control

SecTrust

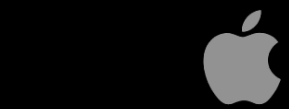
- Purpose
  - Validate certificate
  - Establish user trust for an operation
  - Do this as simply as possible
- SecTrust is a workflow object
  - Create it
  - Add your ingredients
  - Ask it to evaluate

SecTrust (Cont.)

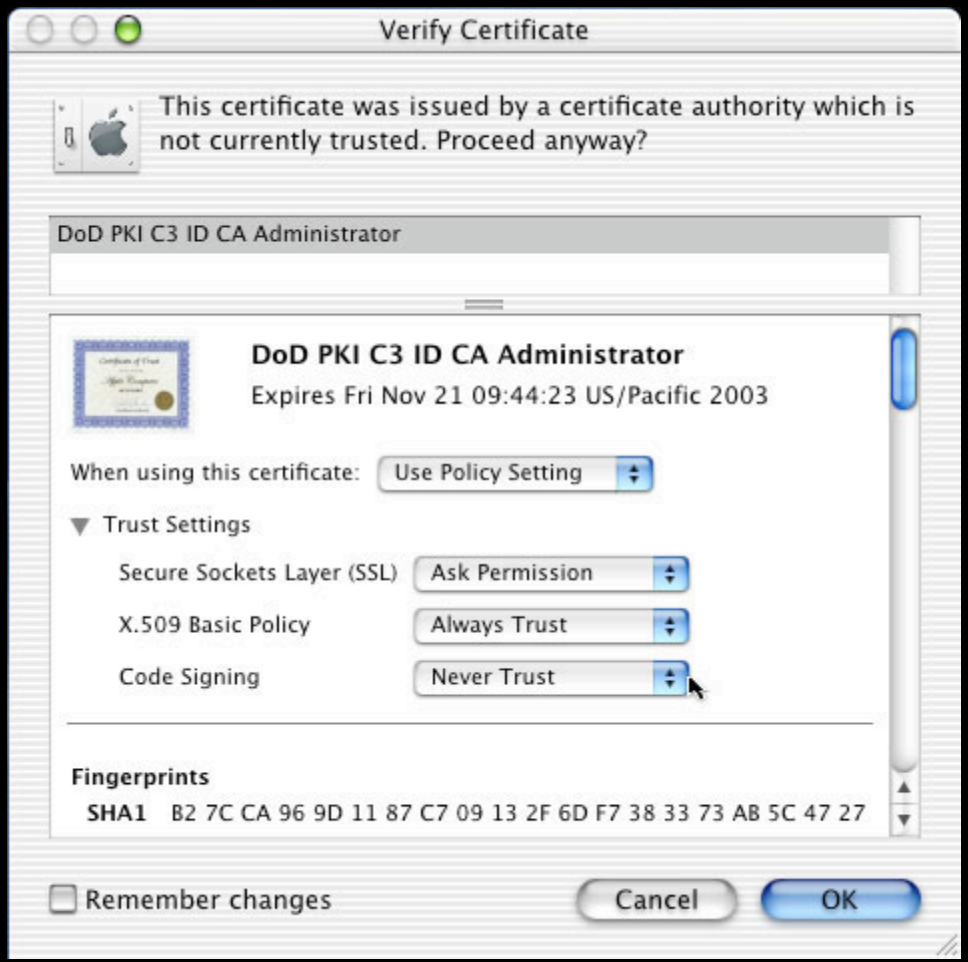
- Outcomes
  - Valid user choice (yes, no, ask, do not know)
  - Recoverable and fatal errors
  - More results on the side
- Ingredients
  - Policies and policy parameters
  - Provide helpful certificates
  - Search keychains for more certificates
  - Anchor certificates

SecUserTrust

- Management API for user trust settings
- Only useful for administrative applications
- UI layer allows user to manage trust directly
- Trust is stored separately by:
    - Certificate
    - Operation
- Trust Values are Yes, No, Ask, and Unset



Editing Certificate Trust



- UI available from Carbon and Cocoa
- Lets users make trust decisions for specific policies

Access Control in CSSM

- Lock-and-key approach: Subjects and Samples
- Attached to keychains, keys, and some items
- Uniform logic: independent of protected object
- ACL maps operations to Subjects
- Owner Subject controls changes to ACL
- Typical Subject/Sample pairs
  - User confirmation
  - Application identity
  - Passphrases

SecAccess

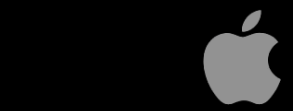
- Encapsulates access control for one item
- Independent of item; transferable
- Make them . . .
  - From an existing item
  - From scratch
- Typical uses
  - Edit an item's access controls
  - Set up initial item access
  - Copy access between items

SecAccess (Cont.)

- SecAccess contains one SecACL per operation
- Simple SecACLs are easy:
  - List of applications (SecTrustedApplication)
  - Item name (for user prompt)
  - User prompt options
- Leave complex ACLs alone

Summary

- New and improved APIs for your pleasure
  - Available in Jaguar
  - For Mac OS X 10.1, stick to Carbon Keychain API
- Easier than CSSM layer (but less flexible)
- Pick what you need, ignore the rest
- Avoid micro-management
- Experts can “bridge down” to CDSA



Security: Certificates in Mac OS X

Craig Keithley
Security and Cryptography Technology Evangelist

Who to Contact

Craig Keithley
Security and Cryptography Technology Evangelist
keithley@apple.com

http://developer.apple.com/wwdc2002/urls.html

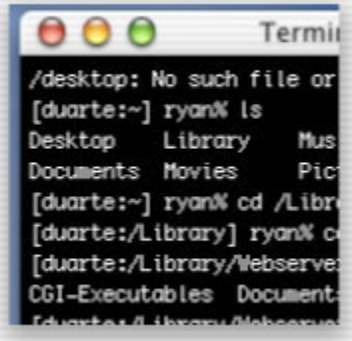
Roadmap

110 Security: Authorization in Mac OS X: Using Authorization Services on OS X	Civic **Wed., 2:00pm**
113 Security: CDSA and Secure Transport: Common Data Security Architecture	Civic **Thurs., 9:00am**
805 Introducing CFNetwork: Communicating with web services	Room C **Tue., 5:00pm**
814 Kerberos in Mac OS X: Learn about Kerberos on Mac OS X	Room C **Thurs., 5:00pm**
FF006 Security: Give us your feedback on security issues	Room J1 **Thurs., 2:00pm**

For More Information

- Apple Developer Security page
  http://developer.apple.com/macos/security.html
- Product Security Web page
  http://support.apple.com/security
- Common Data Security Architecture
  http://opensource.apple.com
  http://www.opengroup.org

Q&A



Craig Keithley
Security and Cryptography Technology Evangelist
keithley@apple.com

http://developer.apple.com/wwdc2002/urls.html


WWDC2002


WWDC2002


WWDC2002